

Bayesian inference for Markov jump processes with informative observations

Andrew Golightly*

Darren J. Wilkinson†

School of Mathematics & Statistics, Newcastle University,
Newcastle-upon-Tyne, NE1 7RU, UK

Abstract

In this paper we consider the problem of parameter inference for Markov jump process (MJP) representations of stochastic kinetic models. Since transition probabilities are intractable for most processes of interest yet forward simulation is straightforward, Bayesian inference typically proceeds through computationally intensive methods such as (particle) MCMC. Such methods ostensibly require the ability to simulate trajectories from the conditioned jump process. When observations are highly informative, use of the forward simulator is likely to be inefficient and may even preclude an exact (simulation based) analysis. We therefore propose three methods for improving the efficiency of simulating conditioned jump processes. A conditioned hazard is derived based on an approximation to the jump process, and used to generate end-point conditioned trajectories for use inside an importance sampling algorithm. We also adapt a recently proposed sequential Monte Carlo scheme to our problem. Essentially, trajectories are reweighted at a set of intermediate time points, with more weight assigned to trajectories that are consistent with the next observation. We consider two implementations of this approach, based on two continuous approximations of the MJP. We compare these constructs for a simple tractable jump process before using them to perform inference for a Lotka-Volterra system. The best performing construct is used to infer the parameters governing a simple model of motility regulation in *Bacillus subtilis*.

Keywords: Bayes; chemical Langevin equation (CLE); linear noise approximation (LNA); Markov jump process (MJP); pMCMC; particle marginal Metropolis-Hastings (PMMH); sequential Monte Carlo (SMC); stochastic kinetic model (SKM).

1 Introduction

Stochastic kinetic models, most naturally represented by Markov jump processes (MJPs), can be used to model a wide range of real-world phenomena including the evolution of biological systems such as intra-cellular processes (Golightly and Wilkinson, 2005; Wilkinson, 2009), predator-prey interaction (Boys et al., 2008; Ferm et al., 2008; Golightly and Wilkinson, 2011) and epidemics (Bailey, 1975; O'Neill and Roberts, 1999; Boys and Giles, 2007). The focus of this paper is to perform exact and fully Bayesian inference for the parameters governing the MJP, using discrete time course observations that may be incomplete and subject to measurement error. A number of recent attempts to address the inference problem have been made. For example, a data augmentation approach was adopted by Boys et al. (2008) and applied to discrete (and error-free) observations of a Lotka-Volterra process. The particle marginal Metropolis-Hastings (PMMH) algorithm

*email: andrew.golightly@ncl.ac.uk

†email: darren.wilkinson@ncl.ac.uk

of Andrieu et al. (2009) has been applied by Golightly and Wilkinson (2011) and Sherlock et al. (2014) to estimate the parameters in model auto-regulatory networks.

The PMMH algorithm offers a promising approach, as it permits a joint update of the parameters and latent process, thus alleviating mixing problems associated with strong correlations. Moreover, the simplest approach is “likelihood-free” in the sense that only forward simulations from the MJP are required. These simulations can be readily obtained by using, for example, Gillespie’s direct method (Gillespie, 1977). The PMMH scheme requires running a sequential Monte Carlo (SMC) scheme (such as the bootstrap particle filter of Gordon et al. (1993)) at every iteration. Given the potential for huge computational burden, improvements to the overall efficiency of PMMH for MJPs has been the focus of Golightly et al. (2014). The latter propose a delayed acceptance analogue of PMMH, (daPMMH), that uses approximations to the MJP such as the chemical Langevin equation (CLE) and linear noise approximation (LNA) (van Kampen, 2001) to screen out parameter draws that are likely to be rejected by the sampler. It should be noted that the simplest likelihood free implementations of both PMMH and daPMMH are likely to perform poorly unless the noise in the measurement error process dominates the intrinsic stochasticity in the MJP. Essentially, in low measurement error cases, only a small number of simulated trajectories will be given reasonable weight inside the SMC scheme, leading to highly variable estimates of marginal likelihood used by the PMMH scheme to construct the acceptance probability. Intolerably long mixing times ensue, unless computational budget permits a large number of particles to be used. In the special case of error-free observation, the algorithm will be impracticable for models of realistic size and complexity, since in this case, trajectories must “hit” the observations.

The development of efficient schemes for generating MJP trajectories that are conditioned on the observations, henceforth referred to as MJP bridges, is therefore of paramount importance. Whilst there is considerable work on the construction of bridges for continuous valued Markov (diffusion) processes (Durham and Gallant, 2002; Delyon and Hu, 2006; Fearnhead, 2008; Stramer and Yan, 2007; Schauer et al., 2014; Del Moral and Murray, 2014), seemingly little has been done for discrete state spaces. The approach taken by Boys et al. (2008) linearly interpolates the hazard function between observation times but requires full and error-free observation of the system of interest. Fan and Shelton (2008) consider an importance sampling algorithm for finite state Markov processes where informative observations are dealt with by sampling reaction times from a truncated exponential distribution and reaction type probabilities are weighted by the probability of reaching the next observation. Hajiaghayi et al. (2014) improve the performance of particle-based Monte Carlo algorithms by analytically marginalising waiting times. The method requires a user-specified potential to push trajectories towards the observation.

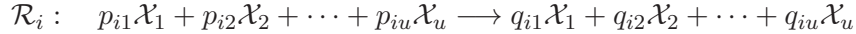
Our novel contribution is an MJP bridge obtained by sampling a jump process with a conditioned hazard that is derived by approximating the expected number of reaction events between observations, given the observations themselves. The resulting hazard is time dependent, however, we find that a simple implementation based on exponential waiting times between proposed reaction events works well in practice. We also adapt the recently proposed bridge particle filter of Del Moral and Murray (2014) to our problem. Their scheme works by generating forward simulations from the process of interest, and reweighting at a set of intermediate times at which resampling may also take place. A look ahead step in the spirit of Lin et al. (2013) prunes out trajectories that are inconsistent with the next observation. The implementation requires an approximation to the (unavailable) transition probability governing the MJP. Del Moral and Murray (2014) used a flexible Gaussian process to approximate the unavailable transition density of a diffusion process. Here, we take advantage of two well known continuous time approximations of the MJP by considering use of the transition density under a discretisation of the CLE or the tractable transition density under the LNA. The methods we propose are simple to implement and are not restricted to finite state spaces.

In section 2, a review of the basic structure of the problem is presented, showing how the Markov

process representation of a reaction network is constructed. In section 3, we consider the problem of sampling conditioned MJPs and give three viable solutions to this problem. In section 4, it is shown how the recently proposed particle MCMC algorithms (Andrieu et al., 2009) may be applied to this class of models. It is also shown how the bridge constructs introduced in section 3 can be used with a pMCMC scheme. The methodology is applied to a number of applications in section 5 before conclusions are drawn in section 6.

2 Stochastic kinetic models

We consider a reaction network involving u species $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_u$ and v reactions $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_v$, with a typical reaction denoted by \mathcal{R}_i and written using standard chemical reaction notation as



Let $X_{j,t}$ denote the number of molecules of species \mathcal{X}_j at time t , and let X_t be the u -vector $X_t = (X_{1,t}, X_{2,t}, \dots, X_{u,t})'$. The dynamics of this model can be described by a vector of rates (or hazards) of the reactions together with a matrix which describes the effect of each reaction on the state. We therefore define a rate function $h_i(X_t, c_i)$, giving the overall hazard of a type i reaction occurring, and we let this depend explicitly on the reaction rate constant c_i , as well as the state of the system at time t . We model the system with a Markov jump process (MJP), so that for an infinitesimal time increment dt , the probability of a type i reaction occurring in the time interval $(t, t + dt]$ is $h_i(X_t, c_i)dt$. When a type i reaction does occur, the system state changes discretely, via the i th row of the so called net effect matrix A , a $v \times u$ matrix with (i, j) th element given by $q_{ij} - p_{ij}$. In what follows, for notational convenience, we work with the stoichiometry matrix defined as $S = A'$. Under the standard assumption of mass action kinetics, the hazard function for a particular reaction of type i takes the form of the rate constant multiplied by a product of binomial coefficients expressing the number of ways in which the reaction can occur, that is,

$$h_i(X_t, c_i) = c_i \prod_{j=1}^u \binom{X_{j,t}}{p_{ij}}.$$

Values for $c = (c_1, c_2, \dots, c_v)'$ and the initial system state $X_0 = x_0$ complete specification of the Markov process. Although this process is rarely analytically tractable for interesting models, it is straightforward to forward-simulate exact realisations of this Markov process using a discrete event simulation method. This is due to the fact that if the current time and state of the system are t and X_t respectively, then the time to the next event will be exponential with rate parameter

$$h_0(X_t, c) = \sum_{i=1}^v h_i(X_t, c_i),$$

and the event will be a reaction of type \mathcal{R}_i with probability $h_i(X_t, c_i)/h_0(X_t, c)$ independently of the waiting time. Forwards simulation of process realisations in this way is typically referred to as *Gillespie's direct method* in the stochastic kinetics literature, after Gillespie (1977). See Wilkinson (2012) for further background on stochastic kinetic modelling.

The primary goal of this paper is that of inference for the stochastic rate constants c , given potentially noisy observations of the system state at a set of discrete times. Golightly and Wilkinson (2011) demonstrated that it is possible to use a particle marginal Metropolis-Hastings (PMMH) scheme for such problems, using only the ability to forward simulate from the system of interest and evaluate the density associated with the observation error process. This ‘‘likelihood free’’ implementation uses the bootstrap particle filter of Gordon et al. (1993). As noted by Golightly and Wilkinson (2011), the efficiency of this algorithm is likely to be very poor when

observations are highly informative. Moreover, in the special case of error-free observation, the algorithm will be computationally intractable for models of realistic size and complexity. We therefore consider three constructions for generating realisations of conditioned jump processes, for use in a PMMH scheme. These constructs rely on the ability to construct both cheap and accurate approximations of the MJP. We therefore consider two candidate approximations in the next section.

2.1 SKM approximations

2.1.1 Chemical Langevin equation

Over an infinitesimal time interval, $(t, t + dt]$, the reaction hazards will remain constant almost surely. The occurrence of reaction events can therefore be regarded as the occurrence of events of a Poisson process with independent realisations for each reaction type. Therefore, the mean and variance of the change in the MJP over the infinitesimal time interval can be calculated as

$$\mathbb{E}(dX_t) = S h(X_t, c)dt, \text{Var}(dX_t) = S \text{diag}\{h(X_t, c)\}S' dt.$$

The Itô stochastic differential equation (SDE) that has the same infinitesimal mean and variance as the true MJP is therefore

$$dX_t = S h(X_t, c)dt + \sqrt{S \text{diag}\{h(X_t, c)\}S'} dW_t, \quad (1)$$

where (without loss of generality) $\sqrt{S \text{diag}\{h(X_t, c)\}S'}$ is a $u \times u$ matrix and W_t is a u -vector of standard Brownian motion. Equation (1) is the SDE most commonly referred to as the chemical Langevin equation (CLE), and can be shown to approximate the SKM increasingly well in high concentration scenarios (Gillespie, 2000). The CLE can rarely be solved analytically, and it is common to work with a discretisation such as the Euler-Maruyama discretisation:

$$\Delta X_t \equiv X_{t+\Delta t} - X_t = S h(X_t, c)\Delta t + \sqrt{S \text{diag}\{h(X_t, c)\}S'} \Delta t Z$$

where Z is a standard multivariate Gaussian random variable.

For a more formal discussion of the CLE and its derivation, we refer the reader to Gillespie (1992) and Gillespie (2000).

2.1.2 Linear noise approximation

The linear noise approximation (LNA) further approximates the MJP by linearising the drift and noise terms of the CLE. The LNA generally possesses a greater degree of numerical and analytic tractability than the CLE. For example, the LNA solution involves (numerically) integrating a set of ODEs for which standard routines, such as the `lsoda` package (Petzold, 1983), exist. The LNA can be derived in a number of more or less formal ways (Kurtz, 1970; Elf and Ehrenberg, 2003; Komorowski et al., 2009). Our brief derivation follows the approach of Wilkinson (2012) to which we refer the reader for further details.

We begin by replacing the hazard function $h(X_t, c)$ in equation (1) with the rescaled form $\Omega f(X_t/\Omega, c)$ where Ω is the volume of the container in which the reactions are taking place. Note that the LNA approximates the CLE increasingly well as Ω and X_t become large, that is, as the system approaches its thermodynamic limit. The CLE then becomes

$$dX_t = \Omega S f(X_t/\Omega, c)dt + \sqrt{\Omega S \text{diag}\{f(X_t/\Omega, c)\}S'} dW_t. \quad (2)$$

We now write the solution X_t of the CLE as a deterministic process plus a residual stochastic process (van Kampen, 2001),

$$X_t = \Omega z_t + \sqrt{\Omega} M_t. \quad (3)$$

We then Taylor expand the rate function around z_t to give

$$f(z_t + M_t/\sqrt{\Omega}, c) = f(z_t, c) + \frac{1}{\sqrt{\Omega}} F_t M_t + O(\Omega^{-1}) \quad (4)$$

where F_t is the $v \times u$ Jacobian matrix with (i, j) th element $\partial f_i(z_t, c)/\partial z_{j,t}$ and we suppress the dependence of F_t on z_t and c for simplicity. Substituting (3) and (4) into equation (2) and collecting terms of $O(1)$ and $O(1/\sqrt{\Omega})$ give the ODE satisfied by z_t , and SDE satisfied by M_t respectively, as

$$dz_t = S f(z_t, c) dt \quad (5)$$

$$dM_t = S F_t M_t dt + \sqrt{S \text{diag}\{f(z_t, c)\} S'} dW_t. \quad (6)$$

Equations (3), (5) and (6) give the linear noise approximation of the CLE and in turn, an approximation of the Markov jump process model.

For fixed or Gaussian initial conditions, that is $M_0 \sim N(m_0, V_0)$, the SDE in (6) can be solved explicitly to give

$$M_t|c \sim N(G_t m_0, G_t \Psi_t G_t')$$

where G_t and Ψ_t satisfy the coupled ODE system given by

$$dG_t = F_t G_t dt; \quad G_0 = I_{u \times u}, \quad (7)$$

$$d\Psi_t = G_t^{-1} S \text{diag}\{f(z_t, c)\} S' (G_t^{-1})'; \quad \Psi_0 = V_0. \quad (8)$$

Hence we obtain

$$X_t \sim N(\Omega z_t + \sqrt{\Omega} G_t m_0, \Omega G_t \Psi_t G_t').$$

In what follows we assume, without loss of generality, that $\Omega = 1$.

3 Sampling conditioned MJPs

We suppose that interest lies in the Markov jump process over an interval $(0, t]$ denoted by $\mathbf{X}(t) = \{X_s | 0 < s \leq t\}$. In fact, it is convenient to denote by $\mathbf{X}(t)$ the collection of reaction times and types over the interval $(0, t]$, which in turn gives the sample path of each species over this interval. Suppose further that the initial state x_0 is a known fixed value and that (a subset of components of) the process is observed at time t subject to Gaussian error, giving a single observation y_t on the random variable

$$Y_t = P' X_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma).$$

Here, Y_t is a length- p vector, P is a constant matrix of dimension $u \times p$ and ε_t is a length- p Gaussian random vector. We denote the density linking Y_t and X_t as $p(y_t|x_t)$. For simplicity, we assume in this section that both Σ and the values of the rate constants c are known, and drop them from the notation where possible.

Our goal is to generate trajectories from $\mathbf{X}(t)|x_0, y_t$ with associated probability function

$$\begin{aligned} \pi(\mathbf{x}(t)|x_0, y_t) &= \frac{p(y_t|x_t)\pi(\mathbf{x}(t)|x_0)}{p(y_t|x_0)} \\ &\propto p(y_t|x_t)\pi(\mathbf{x}(t)|x_0) \end{aligned}$$

where $\pi(\mathbf{x}(t)|x_0)$ is the probability function associated with $\mathbf{x}(t)$. Although $\pi(\mathbf{x}(t)|x_0, y_t)$ will typically be intractable, simulation from $\pi(\mathbf{x}(t)|x_0)$ is straightforward (via Gillespie's direct method), suggesting the construction of a numerical scheme such as Markov chain Monte Carlo or importance sampling. In keeping with the pMCMC methods described in section 4, we focus on the latter.

Algorithm 1 Myopic importance sampling

1. For $i = 1, 2, \dots, N$:
 - (a) Draw $\mathbf{x}(t)^i \sim \pi(\mathbf{x}(t)|x_0)$ using the Gillespie's direct method.
 - (b) Construct the unnormalised weight $\tilde{w}^i = p(y_t|x_t^i)$.
 2. Normalise the weights: $w^i = \tilde{w}^i / \sum_{i=1}^N \tilde{w}^i$.
 3. Resample (with replacement) from the discrete distribution on $\{\mathbf{x}(t)^1, \dots, \mathbf{x}(t)^N\}$ using the normalised weights as probabilities.
-

The simplest importance sampling strategy (given in Algorithm 1) proposes from $\pi(\mathbf{x}(t)|x_0)$ and weights by $p(y_t|x_t)$. If desired, an unweighted sample can easily be obtained by resampling (with replacement) from the discrete distribution over trajectory draws using the normalised weights as probabilities. Plainly, taking the average of the unnormalised weights gives an unbiased estimate of the normalising constant

$$p(y_t|x_0) = \mathbb{E}_{\mathbf{X}(t)|x_0} (p(y_t|X_t)) .$$

This strategy is likely to work well provided that y_t is not particularly informative. The proposal mechanism is independent of the observation y_t and as Σ is reduced, the variance of the importance weights increases. In an error free scenario, with $y_t \equiv x_t$, the unnormalised weights take the value 1 if $x_t^i = x_t$ and are 0 otherwise. Hence, in this extreme scenario, only trajectories that “hit” the observation have non-zero weight.

In order to circumvent these problems, in section 3.1 we derive a novel proposal mechanism based on an approximation of the expected number of reaction events over the interval of interest, conditioned on the observation. In addition, in section 3.2 we adapt a recently proposed bridge particle filter (Del Moral and Murray, 2014) to our problem.

3.1 Conditioned hazard

We suppose that we have simulated as far as time s and derive an approximation of the expected number of reaction events over the interval $(s, t]$. Let ΔR_s denote the number of reaction events over the time $t - s = \Delta s$. We approximate ΔR_s by assuming a constant reaction hazard over Δs . A normal approximation to the corresponding Poisson distribution then gives

$$\Delta R_s \sim \mathcal{N}(h(x_s, c)\Delta s, H(x_s, c)\Delta s)$$

where $H(x_s, c) = \text{diag}\{h(x_s, c)\}$. Under the Gaussian observation regime we have that

$$Y_t|X_s = x_s \sim \mathcal{N}(P'(x_s + S\Delta R_s), P'SH(x_s, c)S'P\Delta s + \Sigma) .$$

Hence, the joint distribution of ΔR_s and Y_t can then be obtained approximately as

$$\begin{pmatrix} \Delta R_s \\ Y_t \end{pmatrix} \sim \mathcal{N} \left\{ \begin{pmatrix} h(x_s, c)\Delta s \\ P'(x_s + S h(x_s, c)\Delta s) \end{pmatrix}, \begin{pmatrix} H(x_s, c)\Delta s & H(x_s, c)S'P\Delta s \\ P'SH(x_s, c)\Delta s & P'SH(x_s, c)S'P\Delta s + \Sigma \end{pmatrix} \right\} .$$

Taking the expectation of $\Delta R_s|Y_t = y_t$ using standard multivariate normal theory, and dividing the resulting expression by Δs gives an approximate conditioned hazard as

$$\begin{aligned} h^*(x_s, c|y_t) &= h(x_s, c) \\ &+ H(x_s, c)S'P(P'SH(x_s, c)S'P\Delta s + \Sigma)^{-1}(y_t - P'[x_s + S h(x_s, c)\Delta s]) . \end{aligned} \quad (9)$$

Algorithm 2 Approximate conditioned MJP generation

1. Set $s = 0$ and $x_s^* = x_0$.
 2. Calculate $h^*(x_s^*, c|y_t)$ and the combined hazard $h_0^*(x_s^*, c|y_t) = \sum_{i=1}^v h_i^*(x_s^*, c_i|y_t)$.
 3. Simulate the time to the next event, $\tau \sim \text{Exp}\{h_0^*(x_s^*, c|y_t)\}$.
 4. Simulate the reaction index, j , as a discrete random quantity with probabilities proportional to $h_i^*(x_s^*, c_i|y_t)$, $i = 1, \dots, v$.
 5. Put $x_{s+\tau}^* = x_s + S^j$ where S^j denotes the j th column of S . Put $s := s + \tau$.
 6. Output x_s^* and s . If $s < t$, return to step 2.
-

A proposed path $\mathbf{x}(t)^*$ can then be produced by sampling reaction events according to an inhomogeneous Poisson process with rate given by (9). An importance sampling scheme based on this proposal mechanism can then be obtained. Although the conditioned hazard in (9) depends on the current time s in a nonlinear way, a simple implementation ignores this time dependence, giving exponential waiting times between proposed reaction events. Algorithm 2 describes the mechanism for generating $\mathbf{x}(t)^*$. To calculate the importance weights, we first note that $\pi(\mathbf{x}(t)|x_0)$ can be written explicitly by considering the generation of all reaction times and types over $(0, t]$. To this end, we let r_j denote the number of reaction events of type \mathcal{R}_j , $j = 1, \dots, v$, and define $n_r = \sum_{j=1}^v r_j$ as the total number of reaction events over the interval. Reaction times (assumed to be in increasing order) and types are denoted by (t_i, ν_i) , $i = 1, \dots, n_r$, $\nu_i \in \{1, \dots, v\}$ and we take $t_0 = 0$ and $t_{n_r+1} = t$. Wilkinson (2012) gives $\pi(\mathbf{x}(t)|x_0)$, also known as the complete data likelihood over $(0, t]$, as

$$\begin{aligned} \pi(\mathbf{x}(t)|x_0) &= \left\{ \prod_{i=1}^{n_r} h_{\nu_i}(x_{t_{i-1}}, c_{\nu_i}) \right\} \exp \left\{ - \sum_{i=1}^{n_r} h_0(x_{t_i}, c) [t_{i+1} - t_i] \right\} \\ &= \left\{ \prod_{i=1}^{n_r} h_{\nu_i}(x_{t_{i-1}}, c_{\nu_i}) \right\} \exp \left\{ - \int_0^t h_0(x_t, c) dt \right\}. \end{aligned}$$

We let $q(\mathbf{x}(t)|x_0, y_t)$ denote the complete data likelihood under the approximate jump process with hazard $h^*(x_s^*, c|y_t)$, so that the importance weight for a path $\mathbf{x}(t)$ is given by

$$\begin{aligned} p(y_t|x_t) \frac{\pi(\mathbf{x}(t)|x_0)}{q(\mathbf{x}(t)|x_0, y_t)} \\ = p(y_t|x_t) \left\{ \prod_{i=1}^{n_r} \frac{h_{\nu_i}(x_{t_{i-1}}, c_{\nu_i})}{h_{\nu_i}^*(x_{t_{i-1}}, c_{\nu_i}|y_t)} \right\} \exp \left\{ - \sum_{i=1}^{n_r} [h_0(x_{t_i}, c) - h_0^*(x_{t_i}, c|y_t)] [t_{i+1} - t_i] \right\}. \end{aligned} \quad (10)$$

When the inhomogeneous Poisson process approximation is sampled exactly, the importance weight in (10) becomes

$$\begin{aligned} p(y_t|x_t) \left\{ \prod_{i=1}^{n_r} \frac{h_{\nu_i}(x_{t_{i-1}}, c_{\nu_i})}{h_{\nu_i}^*(x_{t_{i-1}}, c_{\nu_i}|y_t)} \right\} \exp \left\{ - \int_0^t [h_0(x_t, c) - h_0^*(x_t, c|y_t)] dt \right\} \\ = p(y_t|x_t) \frac{d\mathbb{P}}{d\mathbb{Q}}(\mathbf{x}(t)) \end{aligned}$$

where the last term is seen to be the Radon-Nikodym derivative of the true Markov jump process (\mathbb{P}) with respect to the inhomogeneous Poisson process approximation (\mathbb{Q}) and measures the closeness of the approximating process to the true process.

Algorithm 3 Importance sampling with conditioned hazard

1. For $i = 1, 2, \dots, N$:
 - (a) Draw $\mathbf{x}(t)^i \sim q(\mathbf{x}(t)|x_0, y_t)$ using Algorithm 2.
 - (b) Construct the unnormalised weight

$$\tilde{w}^i = p(y_t|x_t^i) \frac{\pi(\mathbf{x}(t)^i|x_0)}{q(\mathbf{x}(t)^i|x_0, y_t)}$$

whose form is given by (10).

2. Normalise the weights: $w^i = \tilde{w}^i / \sum_{i=1}^N \tilde{w}^i$.
 3. Resample (with replacement) from the discrete distribution on $\{\mathbf{x}(t)^1, \dots, \mathbf{x}(t)^N\}$ using the normalised weights as probabilities.
-

Algorithm 3 gives importance sampling algorithm that uses an approximate implementation of the inhomogeneous Poisson process approximation. Note that in the special case of no error, the importance weight in step 1(b) has $p(y_t|x_t^i)$ replaced with an indicator function assigning the value 1 if $x_t^i = x_t$ and 0 otherwise. Upon completion of the algorithm, an equally weighted sample approximately distributed according to $\pi(\mathbf{x}(t)|x_0, y_t)$ is obtained. The average unnormalised weight can be used to (unbiasedly) estimate the normalising constant $p(y_t|x_0)$.

3.2 Bridge particle filter

Del Moral and Murray (2014) considered the problem of sampling continuous time, continuous valued Markov processes and proposed a bridge particle filter to weight forward trajectories based on an approximation to the unknown transition probabilities at each reweighting step. Here, we adapt their method to our problem. We note that when using the bridge particle filter to sample MJP trajectories, it is possible to obtain a likelihood free scheme.

Without loss of generality, we adopt an equispaced partition of $[0, t]$ as

$$0 = t_0 < t_1 < \dots < t_n = t.$$

This partition is used to determine the times at which resampling may take place. Introduce the weight functions

$$W_k(x_{t_{k-1}:t_k}) = \frac{q(y_t|x_{t_k})}{q(y_t|x_{t_{k-1}})}$$

where $q(y_t|x_{t_k})$, $k = 0, \dots, n$ are positive functions. Note that

$$\frac{q(y_t|x_0)}{q(y_t|x_t)} \prod_{k=1}^n W_k(x_{t_{k-1}:t_k}) = 1$$

and write $\pi(\mathbf{x}(t)|x_0, y_t)$ as

$$\begin{aligned}
\pi(\mathbf{x}(t)|x_0, y_t) &\propto p(y_t|x_t)\pi(\mathbf{x}(t)|x_0)\frac{q(y_t|x_0)}{q(y_t|x_t)}\prod_{k=1}^n W_k(x_{t_{k-1}:t_k}) \\
&\propto p(y_t|x_t)\frac{q(y_t|x_0)}{q(y_t|x_t)}\prod_{k=1}^n W_k(x_{t_{k-1}:t_k})\pi(\mathbf{x}(t_{k-1}:t_k)|x_{t_{k-1}}) \\
&\propto \prod_{k=1}^n W_k(x_{t_{k-1}:t_k})\pi(\mathbf{x}(t_{k-1}:t_k)|x_{t_{k-1}})
\end{aligned} \tag{11}$$

where $\pi(\mathbf{x}(t_{k-1}:t_k)|x_{t_{k-1}})$ denotes the probability function associated with $\mathbf{X}(t_{k-1}:t_k) = \{X_s | t_{k-1} < s \leq t_k\}$ and the last line (11) follows by taking $q(y_t|x_t)$ to be $p(y_t|x_t)$ and absorbing $q(y_t|x_0)$ into the proportionality constant. The form of (11) suggests a sequential Monte Carlo (SMC) scheme (also known as a particle filter) where at time t_{k-1} each particle (trajectory) $\mathbf{x}(t_{k-1})^i$ is extended by simulating from $\pi(\mathbf{x}(t_{k-1}:t_k)|x_{t_{k-1}}^i)$ and incrementally weighted by $W_k(x_{t_{k-1}:t_k})$. Intuitively, by “looking ahead” to the observation, trajectories that are not consistent with y_t are given small weight and should be pruned out with a resampling step. Del Moral and Murray (2014) suggest an adaptive resampling procedure so that resampling is only performed if the effective sample size (ESS) falls below some fraction of the number of particles, say β . The ESS is defined (Liu and Chen, 1995) as a function of the weights $w^{1:N}$ by

$$ESS(w^{1:N}) = \frac{\left(\sum_{i=1}^N w^i\right)^2}{\sum_{i=1}^N (w^i)^2}. \tag{12}$$

It remains that we can choose sensible functions $q(y_t|x_{t_k})$ to be used to construct the weights. We propose to use the density associated with $Y_t|X_{t_k} = x_{t_k}$ under the CLE or LNA:

$$\begin{aligned}
q_{CLE}(y_t|x_{t_k}) &= N(y_t; P'[x_{t_k} + S h(x_{t_k}, c)(t - t_k)], P' S H(x_{t_k}, c) S' P(t - t_k) + \Sigma), \\
q_{LNA}(y_t|x_{t_k}) &= N(y_t; P'[z_t + G_{t-t_k}(x_{t_k} - z_{t_k})], P' G_{t-t_k} \Psi_{t-t_k} G_{t-t_k}' P + \Sigma).
\end{aligned}$$

Note that due to the intractability of the CLE, we propose to use a single step of the Euler-Mauryama approximation. Comments on the relative merits of each scheme are given in Section 3.3.

Algorithm 4 gives the sequence of steps necessary to implement the bridge particle filter. The average unnormalised weight obtained at time t can be used to estimate the normalising constant $p(y_t|x_0)$:

$$\hat{p}(y_t|x_0) \propto \frac{1}{N} \sum_{i=1}^N \tilde{w}_n^i.$$

We now consider some special cases of Algorithm 4. For unknown x_0 with prior probability mass function $\pi(x_0)$, the target becomes

$$\pi(\mathbf{x}(t), x_0|y_t) \propto \pi(x_0)q(y_t|x_0)\prod_{k=1}^n W_k(x_{t_{k-1}:t_k})\pi(\mathbf{x}(t_{k-1}:t_k)|x_{t_{k-1}})$$

which suggests that step 1(a) should be replaced by sampling particles $x_0^i \sim \pi(x_0)$. The contribution $q(y_t|x_0)$ could either be absorbed into the final weight (taking care to respect the ancestral lineage of the trajectory), or an initial look ahead step could be performed by resampling amongst the x_0^i with weights proportional to $q(y_t|x_0^i)$. If the latter strategy is adopted and no additional resampling steps are performed, the algorithm reduces to the auxiliary particle filter of Pitt and Shephard (1999), where particles are pre-weighted by $q(y_t|x_0)$ and propagated through myopic forward simulation.

Algorithm 4 Bridge particle filter

1. Initialise. For $i = 1, 2, \dots, N$:
 - (a) Set $x_0^i = x_0$ and put $w_0^i = 1/N$.
 2. Perform sequential importance sampling. For $k = 1, 2, \dots, n$ and $i = 1, 2, \dots, N$:
 - (a) If $ESS(w_{k-1}^{1:N}) < \beta N$ draw indices a_k^i from the discrete distribution on $\{1, \dots, N\}$ with probabilities given by $w_{k-1}^{1:N}$ and put $w_k^i = 1/N$. Otherwise, put $a_k^i = i$.
 - (b) Draw $\mathbf{x}(t_{k-1} : t_k)^i \sim \pi(\cdot | x_{t_{k-1}}^{a_k^i})$ using the Gillespie algorithm initialised at $x_{t_{k-1}}^{a_k^i}$.
 - (c) Construct the unnormalised weight

$$\tilde{w}_k^i = \tilde{w}_{k-1}^i \frac{q(y_t | x_{t_k}^i)}{q(y_t | x_{t_{k-1}}^{a_k^i})}$$
 - (d) Normalise the weights: $w_k^i = \tilde{w}_k^i / \sum_{i=1}^N \tilde{w}_k^i$.
 3. Let $b_n^i = i$ and define $b_k^i = a_{k+1}^{b_k^i}$ recursively. Resample (with replacement) from the discrete distribution on $\{(\mathbf{x}(0 : t_1)^{b_1^i}, \dots, \mathbf{x}(t_{n-1} : t)^i), i = 1, \dots, N\}$ using the normalised weights as probabilities.
-

If no resampling steps are performed at any time, the algorithm reduces to the myopic importance sampling strategy described in Section 1.

In the error free scenario, the target can be written as

$$\pi(\mathbf{x}(t) | x_t, x_0) \propto \frac{q(x_t | x_0)}{q(x_t | x_{t_{n-1}})} \pi(\mathbf{x}(t_{n-1} : t) | x_{t_{n-1}}) \prod_{k=1}^{n-1} W_k(x_{t_{k-1}:t_k}) \pi(\mathbf{x}(t_{k-1} : t_k) | x_{t_{k-1}})$$

where the incremental weight functions are redefined as

$$W_k(x_{t_{k-1}:t_k}) = \frac{q(x_t | x_{t_k})}{q(x_t | x_{t_{k-1}})}.$$

The form of the target suggests that at time t_{n-1} , particle trajectories should be propagated via $\pi(\mathbf{x}(t_{n-1} : t) | x_{t_{n-1}})$ and weighted by $q(x_t | x_0) / q(x_t | x_{t_{n-1}})$, provided that the trajectory “hits” the observation x_t , otherwise a weight of 0 should be assigned. Hence, unlike in the continuous state space scenario considered by Del Moral and Murray (2014), the algorithm is likelihood free, in the sense that $\pi(\mathbf{x}(t_{n-1} : t) | x_{t_{n-1}})$ need not be evaluated.

3.3 Comments on efficiency

Application of Algorithm 3 requires calculation of the conditioned hazard function in (9) after every reaction event. The cost of this calculation will therefore be dictated by the number of observed components p , given that a $p \times p$ matrix must be inverted. Despite this, for many systems of interest, it is unlikely that all components will be observed and we anticipate that in practice $p \ll u$, where u is the number of species in the system. The construction of the conditioned hazard is based on an assumption that the hazard function is constant over diminishing time intervals $(s, t]$ and that the number of reactions over this interval is approximately Gaussian. The performance of the construct is therefore likely to diminish if applied over time horizons during which the reaction hazards vary

substantially. We also note that the elements of the conditioned hazard are not guaranteed to be positive and we therefore truncate each hazard component at zero.

We implement the bridge particle filter in Algorithm 4 with the weight functions obtained either through the CLE or the LNA. To maximise statistical efficiency, we require that $q(y_t|x_{t_k}) \approx p(y_t|x_{t_k})$. Given the analytic intractability of the CLE, we obtain q via a single step of an Euler-Maruyama scheme. Whilst this is likely to be computationally efficient, given the simplicity of applying a single step of the Euler-Maruyama scheme, we anticipate that applying the scheme over large time intervals (where non-linear dynamics are observed) is likely to be unsatisfactory. The tractability of the LNA has been recently exploited (Komorowski et al., 2009; Fearnhead et al., 2014; Golightly et al., 2014) and shown to give a reasonable approximation to the MJP for a number of reaction networks. However, use of the LNA requires the solution of a system of $u(u+1)/2 + 2u$ coupled ODEs. For most stochastic kinetic models of interest, the solution to the LNA ODEs will not be analytically tractable. Whilst good numerical ODE solvers are readily available, the bridge particle filter is likely to require a full numerical solution over the time interval of interest for each particle (except in the special error free case where only a single solution is required). Both the CLE and LNA replace the intractable transition probability with a Gaussian approximation. Moreover, the approximations may be light tailed relative to the target, and erstwhile valid trajectories may be pruned out by the resampling procedure. Tempering the approximations by raising $q(y_t|x_{t_k})$ to a power γ ($0 < \gamma < 1$) may alleviate this problem at the expense of choosing an appropriate value for the additional tuning parameter γ . We assess the empirical performance of each scheme in Section 5.

4 Bayesian inference

Consider a time interval $[0, T]$ over which a Markov jump process $\mathbf{X} = \{X_t | 0 \leq t \leq T\}$ is not observed directly, but observations (on a regular grid) $\mathbf{y} = \{y_t | t = 0, 1, \dots, T\}$ are available and assumed conditionally independent (given \mathbf{X}) with conditional probability distribution obtained via the observation equation,

$$Y_t = P'X_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma), \quad t = 0, 1, \dots, T. \quad (13)$$

As in Section 3, we take Y_t to be a length- p vector, P is a constant matrix of dimension $u \times p$ and ε_t is a length- p Gaussian random vector. We assume that primary interest lies in the rate constants c where, in the case of unknown measurement error variance, the parameter vector c is augmented to include the parameters of Σ . Bayesian inference may then proceed through the marginal posterior density

$$p(c|\mathbf{y}) \propto p(c)p(\mathbf{y}|c) \quad (14)$$

where $p(c)$ is the prior density ascribed to c and $p(\mathbf{y}|c)$ is the marginal likelihood. Since the posterior in (14) will be intractable in practice, samples are usually generated from (14) via MCMC. A further complication is the intractability of the marginal likelihood term, and we therefore adopt the particle marginal Metropolis-Hastings scheme of Andrieu et al. (2010) which has been successfully applied to stochastic kinetic models in Golightly and Wilkinson (2011) and Golightly et al. (2014).

4.1 Particle marginal Metropolis-Hastings

Since interest lies in the marginal posterior in (14), we consider the special case of the particle marginal Metropolis-Hastings (PMMH) algorithm (Andrieu et al., 2010) which can be seen as a pseudo-marginal MH scheme (Beaumont, 2003; Andrieu and Roberts, 2009). Under some fairly mild conditions (for which we refer the reader to Del Moral (2004) and Andrieu et al. (2010)), a sequential Monte Carlo scheme targeting the probability associated with the conditioned MJP,

$\pi(\mathbf{x}|\mathbf{y}, c)$, can be implemented to give an unbiased estimate of the marginal likelihood. We write this estimate as $\hat{p}(\mathbf{y}|c, u)$ where u denotes all random variables generated by the SMC scheme according to some density $q(u|c)$. We now consider a target of the form

$$\hat{p}(c, u|\mathbf{y}) \propto \hat{p}(\mathbf{y}|c, u)q(u|c)p(c)$$

for which marginalisation over u gives

$$\begin{aligned} \int \hat{p}(c, u|\mathbf{y}) du &\propto p(c) \mathbb{E}_{u|c} \{ \hat{p}(\mathbf{y}|c, u) \} \\ &\propto p(c)p(\mathbf{y}|c). \end{aligned}$$

Hence, a MH scheme targeting $\hat{p}(c, u|\mathbf{y})$ with proposal kernel $q(c^*|c)q(u^*|c^*)$ accepts a move from (c, u) to (c^*, u^*) with probability

$$\frac{\hat{p}(\mathbf{y}|c^*, u^*)p(c^*)}{\hat{p}(\mathbf{y}|c, u)p(c)} \times \frac{q(c|c^*)}{q(c^*|c)}.$$

We see that the values of u need never be stored and it should now be clear that the scheme targets the correct marginal distribution $p(c|\mathbf{y})$.

4.2 Implementation

Algorithms 3 and 4 can readily be applied to give an SMC scheme targeting $\pi(\mathbf{x}|\mathbf{y}, c)$. In each case, an initialisation step should be performed where a weighted sample $\{(x_0^i, w_0^i), i = 1, \dots, N\}$ is obtained by drawing values x_0^i from some prior with mass function $\pi(x_0)$ and assigning weights proportional to $p(y_0|x_0^i, c)$. If desired, resampling could be performed so that the algorithm is initialised with an equally weighted sample drawn from $\pi(x_0|y_0, c)$. Algorithms 3 and 4 can then be applied sequentially, for times $t = 1, 2, \dots, T$, simply by replacing x_0 with x_{t-1}^i . After assimilating all information, an unbiased estimate of the marginal likelihood $p(\mathbf{y}|c)$ is obtained as

$$\hat{p}(\mathbf{y}|c) = \hat{p}(y_0|c) \prod_{t=1}^T \hat{p}(y_t|\mathbf{y}(t-1)) \quad (15)$$

where $\mathbf{y}(t-1) = \{y_t, t = 0, 1, \dots, t-1\}$ and we have dropped u from the notation for simplicity. The product in (15) can be obtained from the output of Algorithms 3 and 4. For example, when using the conditioned hazard approach, (15) is simply the product of the average unnormalised weight obtained in step 1(b). Use of Algorithms 3 and 4 in this way give SMC schemes that fall into a class of auxiliary particle filters (Pitt and Shephard, 1999). We refer the reader to Pitt et al. (2012) for a theoretical treatment of the use of an auxiliary particle filter inside an MH scheme.

The mixing of the PMMH scheme is likely to depend on the number of particles used in the SMC scheme. Whilst the method can be implemented using just $N = 1$ particle, the corresponding estimator of marginal likelihood will be highly variable, and the impact of this on the PMMH algorithm will be a poorly mixing chain. As noted by Andrieu and Roberts (2009), the mixing efficiency of the PMMH scheme decreases as the variance of the estimated marginal likelihood increases. This problem can be alleviated at the expense of greater computational cost by increasing N . This therefore suggests an optimal value of N and finding this choice is the subject of Sherlock et al. (2013) and Doucet et al. (2014). The former show that for a “standard asymptotic regime” N should be chosen so that the variance in the noise in the estimated log-posterior is around 3, but find that for low dimensional problems a smaller value (around 2) is optimal. We therefore recommend performing an initial pilot run of PMMH to obtain an estimate of the posterior mean (or median) parameter value, and a (small) number of additional sampled values. The value of N should then

be chosen so that the variance of the noise in the estimated log-posterior is (ideally) in the range $[2, 4]$.

Since all parameter values must be strictly positive we adopt a proposal kernel corresponding to a random walk on $\log(c)$, with Gaussian innovations. We take the innovation variance to be $\lambda \widehat{\text{Var}}(\log(c))$ and follow the practical advice of Sherlock et al. (2013) by tuning λ to give an acceptance rate of around 15%.

5 Applications

In order to examine the empirical performance of the methods proposed in section 3, we consider three examples. These are a simple (and tractable) birth-death model, the stochastic Lotka-Volterra model examined by Boys et al. (2008) and a systems biology model of bacterial motility regulation (Wilkinson, 2011).

5.1 Birth-Death

The birth-death reaction network takes the form

$$\mathcal{R}_1 : \mathcal{X}_1 \longrightarrow 2\mathcal{X}_1, \quad \mathcal{R}_2 : \mathcal{X}_1 \longrightarrow \emptyset$$

with birth and death reactions shown respectively. The stoichiometry matrix is given by

$$S = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(x_t, c) = (c_1 x_t, c_2 x_t)'$$

where x_t denotes the state of the system at time t . The CLE is given by

$$dX_t = (c_1 - c_2) X_t dt + \sqrt{(c_1 + c_2) X_t} dW_t$$

which can be seen as a degenerate case of a Feller square-root diffusion (Feller, 1952). For reaction networks of reasonable size and complexity, the CLE will be intractable. To explore the effect of working with a numerical approximation of the CLE inside the bridge particle filter, we adopt the Euler-Maruyama approximation which gives (for a fixed initial condition x_0) an approximation to the transition density as

$$X_t | X_0 = x_0 \sim N(x_0 + (c_1 - c_2)x_0 t, (c_1 + c_2)x_0 t).$$

The ODE system governing the LNA with initial conditions $z_0 = x_0$, $m_0 = 0$ and $V_0 = 0$ can be solved analytically to give

$$X_t | X_0 = x_0 \sim N\left(x_0 e^{(c_1 - c_2)t}, x_0 \frac{(c_1 + c_2)}{(c_1 - c_2)} e^{(c_1 - c_2)t} \left[e^{(c_1 - c_2)t} - 1\right]\right).$$

We consider an example in which $c = (0.5, 1)$ and $x_0 = 100$ are fixed. To provide a challenging scenario we took x_t to be the upper 99% quantile of $X_t | X_0 = 100$. To assess the performance of each algorithm as an observation is made with increasing time sparsity, we took $t \in \{0.1, 0.5, 1\}$. Algorithms 1 (denoted MIS), 3 (denoted CH) and 4 (denoted BPF-CLE or BPF-LNA) were run with $N \in \{10, 50, 100, 500\}$ to give a set of $m = 5000$ estimates of the transition probability $\pi(x_t | x_0)$ and we denote this set by $\hat{\pi}_N^{1:m}(x_t | x_0)$. The bridge particle filter also requires specification of the intermediate time points at which resampling could take place. For simplicity, we took an

| Method | N | $t = 0.1$ | $t = 0.5$ | $t = 1$ |
|---------|-----|----------------------------------|----------------------------------|----------------------------------|
| MIS | 10 | 300, 293, 6.2×10^{-4} | 171, 168, 3.5×10^{-4} | 151, 149, 3.0×10^{-4} |
| | 50 | 1340, 1190, 1.2×10^{-4} | 827, 773, 7.0×10^{-5} | 682, 639, 5.8×10^{-5} |
| | 100 | 2331, 1921, 6.4×10^{-5} | 1488, 1308, 3.5×10^{-5} | 1364, 1203, 3.2×10^{-5} |
| | 500 | 4776, 3771, 1.2×10^{-5} | 4196, 3230, 6.8×10^{-6} | 3901, 3004, 6.1×10^{-6} |
| CH | 10 | 4974, 3264, 1.6×10^{-5} | 4985, 2998, 7.8×10^{-6} | 4990, 3581, 2.4×10^{-6} |
| | 50 | 5000, 4395, 4.6×10^{-6} | 5000, 4546, 1.2×10^{-6} | 5000, 4508, 9.7×10^{-7} |
| | 100 | 5000, 4689, 2.4×10^{-6} | 5000, 4668, 8.5×10^{-7} | 5000, 4798, 3.8×10^{-7} |
| | 500 | 5000, 4921, 7.7×10^{-7} | 5000, 4943, 1.6×10^{-7} | 5000, 4939, 1.2×10^{-7} |
| BPF-CLE | 10 | 2581, 349, 5.7×10^{-4} | 2412, 556, 7.7×10^{-5} | 2745, 532, 1.7×10^{-5} |
| | 50 | 4982, 2137, 6.3×10^{-5} | 4920, 3391, 4.9×10^{-6} | 3236, 4925, 4.0×10^{-6} |
| | 100 | 5000, 3519, 1.9×10^{-5} | 4998, 3979, 2.8×10^{-6} | 4999, 4106, 4.1×10^{-6} |
| | 500 | 5000, 3841, 1.5×10^{-5} | 5000, 4756, 6.7×10^{-7} | 5000, 4780, 3.2×10^{-6} |
| BPF-LNA | 10 | 2634, 403, 4.3×10^{-4} | 2514, 636, 6.9×10^{-5} | 2843, 1102, 2.4×10^{-5} |
| | 50 | 4963, 2748, 3.2×10^{-5} | 4926, 3198, 6.0×10^{-6} | 4949, 3625, 2.8×10^{-6} |
| | 100 | 5000, 3612, 1.5×10^{-5} | 4998, 4055, 2.5×10^{-6} | 5000, 4016, 1.9×10^{-6} |
| | 500 | 5000, 3643, 1.4×10^{-5} | 5000, 4655, 8.8×10^{-7} | 5000, 4771, 5.4×10^{-7} |

Table 1: $\sum_{i=1}^m I(\hat{\pi}_N(x_t|x_0) > 0)$, $\text{ESS}(\hat{\pi}_N^{1:m}(x_t|x_0))$ and $\text{MSE}(\hat{\pi}_N^{1:m}(x_t|x_0))$, based on 5000 runs of MIS, CH, BPF-CLE and BPF-LNA. For MIS, the expected number of non-zero estimates (as obtained analytically) is reported. In all cases, $x_0 = 100$ and x_t is the upper 99% quantile of $X_t|X_0 = 100$.

equispaced partition of $[0, t]$ with a time step of 0.02 for $t = 0.1$, and 0.05 for $t \in \{0.5, 1\}$. We found that these gave a good balance between statistical efficiency and CPU time.

To compare the algorithms, we report the number of non-zero normalising constant estimates $\sum_{i=1}^m I(\hat{\pi}_N(x_t|x_0) > 0)$, the effective sample size $\text{ESS}(\hat{\pi}_N^{1:m}(x_t|x_0))$ whose form is defined in (12) and mean-squared error $\text{MSE}(\hat{\pi}_N^{1:m}(x_t|x_0))$ given by

$$\text{MSE}(\hat{\pi}_N^{1:m}(x_t|x_0)) = \frac{1}{m} \sum_{i=1}^m [\hat{\pi}_N^i(x_t|x_0) - \pi(x_t|x_0)]^2$$

where $\pi(x_t|x_0)$ can be obtained analytically (Bailey, 1964).

The results are summarised in Table 1. Use of the conditioned hazard and bridge particle filters (CH, BPF-CLE and BPF-LNA) comprehensively outperform the myopic importance sampler (MIS). For example, for the $t = 1$ case, an order of magnitude improvement is observed when comparing BPF (CLE or LNA) with MIS in terms of mean squared error. We see a reduction in mean squared error of two orders of magnitude when comparing MIS with CH, across all experiments, and performance (across all metrics) of MIS with $N = 500$ is comparable with the performance of CH when $N = 10$. BPF-LNA generally outperforms BPF-CLE, although the difference is small. Running the BPF schemes generally requires twice as much computational effort than MIS, whereas CH is roughly three times slower than MIS. Even when this additional cost is taken into account, MIS cannot be recommended in this example.

Naturally, the performance of BPF will depend on the accuracy of the normal approximations used by the CLE and LNA. In particular, we expect these approximations to be unsatisfactory when species numbers are low. Moreover, when the conditioned jump process exhibits nonlinear dynamics, we expect the Euler approximation to be particularly poor. We therefore repeated the experiments of Table 1 with $N = 500$, $x_0 = 10$, and x_t as the lower 1% quantile of $X_t|X_0 = 10$. Results are reported in Table 2. We see that in this case, MIS outperforms BPF and the performance of BPF-CLE worsens as t increases, suggesting that a single step of the Euler approximation is

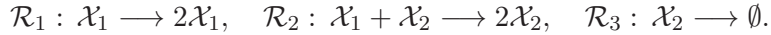
| Method | $t = 0.1$ | $t = 0.5$ | $t = 1$ |
|---------|----------------------------------|----------------------------------|-----------------------------------|
| MIS | 5000, 4747, 7.3×10^{-5} | 4998, 4426, 3.1×10^{-5} | 4999, 4500, 3.7×10^{-5} |
| CH | 5000, 4979, 8.7×10^{-6} | 5000, 4963, 2.3×10^{-6} | 5000, 4965, 2.58×10^{-6} |
| BPF-CLE | 5000, 4131, 3.9×10^{-4} | 5000, 3013, 1.4×10^{-3} | 5000, 3478, 1.8×10^{-3} |
| BPF-LNA | 5000, 3946, 3.6×10^{-4} | 5000, 3667, 1.4×10^{-4} | 5000, 3639, 1.3×10^{-4} |

Table 2: $\sum_{i=1}^m I(\hat{\pi}_N(x_t|x_0) > 0)$, $\text{ESS}(\hat{\pi}_N^{1:m}(x_t|x_0))$ and $\text{MSE}(\hat{\pi}_N^{1:m}(x_t|x_0))$, based on 5000 runs of MIS, CH, BPF-CLE and BPF-LNA. For MIS, the expected number of non-zero estimates (as obtained analytically) is reported. In all cases, $N = 500$, $x_0 = 10$ and x_t is the lower 1% quantile of $X_t|X_0 = 10$.

unsatisfactory for $t > 0.1$. Use of the conditioned hazard on the other hand appears fairly robust to different choices of x_0 , x_t and t .

5.2 Lotka-Volterra

We consider a simple model of predator and prey interaction comprising three reactions:



Denote the current state of the system by $X = (X_1, X_2)'$ where we have dropped dependence of the state on t for notational simplicity. The stoichiometry matrix is given by

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X, c) = (c_1 X_1, c_2 X_1 X_2, c_3 X_2)'$$

We consider three synthetic datasets consisting of 51 observations at integer times on prey and predator levels generated from the stochastic kinetic model using Gillespie's direct method and corrupted with zero mean Gaussian noise. The observation equation (13) is therefore

$$Y_t = X_t + \varepsilon_t,$$

where $X_t = (X_{1,t}, X_{2,t})'$, $\varepsilon_t \sim \text{N}(0, \sigma^2)$. We took $\sigma = 10$ to construct the first dataset (\mathcal{D}_1), $\sigma = 5$ to construct the second (\mathcal{D}_2) and $\sigma = 1$ to give the third synthetic dataset (\mathcal{D}_3). In all cases we assumed σ^2 to be known. True values of the rate constants $(c_1, c_2, c_3)'$ were taken to be 0.5, 0.0025, and 0.3 following Boys et al. (2008). We took the initial latent state as $x_0 = (71, 79)'$ assumed known for simplicity. Independent proper Uniform $U(-8, 8)$ priors were ascribed to each $\log(c_i)$, denoted by θ_i , $i = 1, 2, 3$ and we let $\theta = (\theta_1, \theta_2, \theta_3)'$ be the quantity for which inferences are to be made.

For brevity, we refer to the likelihood-free PMMH scheme (based on forward simulation only) as PMMH-LF, and the scheme based on the conditioned hazard proposal mechanism as PMMH-CH. As the ODEs governing the LNA solution are intractable, we focus on the CLE implementation of the bridge particle filter and refer to this scheme as PMMH-BPF. A pilot run of PMMH-LF was performed for each dataset to give an estimate of the posterior variance $\widehat{\text{Var}}(\theta)$, posterior median and 3 additional sampled θ values. We denote the variance of the noise in the log posterior by τ^2 and chose the number of particles N for each scheme so that $\tau^2 \approx 2$ at the estimated posterior median and $\tau^2 < 4$ at the remaining sampled θ values (where possible). We updated θ using a Gaussian random walk with an innovation variance given by $\lambda \widehat{\text{Var}}(\theta)$, with the scaling parameter λ optimised

| | N | τ^2 | Acc. rate | ESS($\theta_1, \theta_2, \theta_3$) | Time (s) | ESS _{min} /s |
|-----------------------------------|-------|----------|-----------|---------------------------------------|----------|-----------------------|
| \mathcal{D}_1 ($\sigma = 10$) | | | | | | |
| PMMH-LF | 230 | 2.0 | 0.15 | (3471, 3465, 3760) | 17661 | 0.196 |
| PMMH-CH | 50 | 2.1 | 0.14 | (3178, 3153, 3095) | 18773 | 0.165 |
| PMMH-BPF | 220 | 2.0 | 0.16 | (3215, 2994, 3121) | 27874 | 0.107 |
| \mathcal{D}_2 ($\sigma = 5$) | | | | | | |
| PMMH-LF | 440 | 2.0 | 0.15 | (3482, 3845, 3784) | 33808 | 0.103 |
| PMMH-CH | 35 | 2.0 | 0.15 | (3581, 3210, 3204) | 13341 | 0.240 |
| PMMH-BPF | 250 | 1.9 | 0.17 | (3779, 3887, 4110) | 33436 | 0.113 |
| \mathcal{D}_3 ($\sigma = 1$) | | | | | | |
| PMMH-LF | 25000 | 1.9 | 0.18 | (2503, 2746, 2472) | 1277834 | 0.00193 |
| PMMH-CH | 55 | 1.9 | 0.14 | (2861, 2720, 2844) | 22910 | 0.118 |
| PMMH-BPF | 3000 | 1.8 | 0.18 | (3732, 3990, 4221) | 290000 | 0.0129 |

Table 3: Lotka-Volterra model. Number of particles N , variance of the noise in the log-posterior (τ^2) at the posterior median, acceptance rate, effective sample size (ESS) of each parameter chain and wall clock time in seconds and minimum (over each parameter chain) ESS per second.

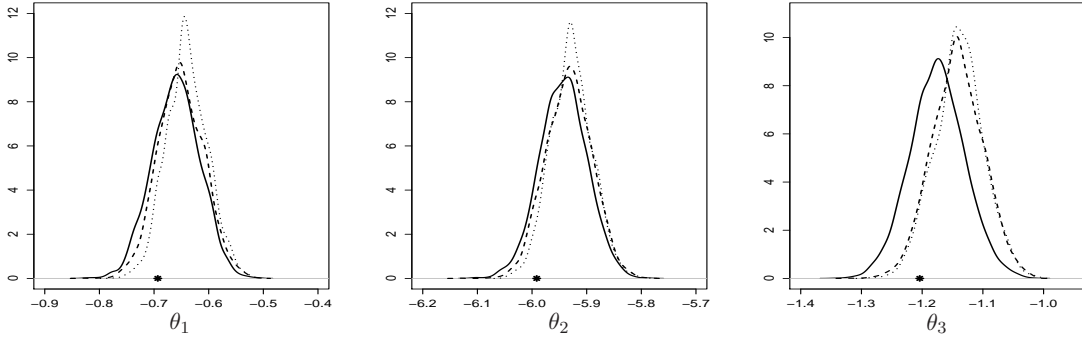


Figure 1: Lotka-Volterra model. Marginal posterior distributions based on synthetic data generated using $\sigma^2 = 10$ (solid), $\sigma^2 = 5$ (dashed) and $\sigma^2 = 1$ (dotted). Values of each θ_i that produced the data are indicated.

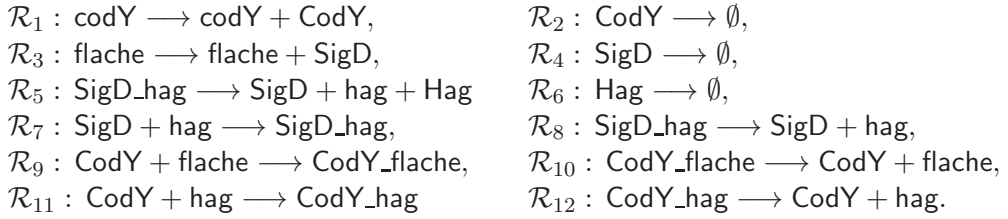
empirically, using minimum effective sample size (ESS_{min}) over each parameter chain. PMMH-BPF requires specification of a set of intermediate times at which resampling could be triggered. We found that resampling every 0.2 time units worked well. We also found that tempering the CLE approximation by raising each contribution $q(y_t|x_{t_k})$ to the power γ performed better than using the CLE approximation directly (with $\gamma = 1$). We took $\gamma = 0.5, 0.2$ and 0.1 for each dataset \mathcal{D}_1 , \mathcal{D}_2 and \mathcal{D}_3 respectively. All schemes were run for 10^5 iterations, except for PMMH-LF when using dataset \mathcal{D}_1 , whose computational cost necessitated a shorter run of 50,000 iterations. All algorithms are coded in C and run on a desktop computer with a 3.4GHz clock speed.

Figure 1 shows the marginal posterior distributions for each dataset and Table 3 summarises the overall efficiency of each PMMH scheme. When using PMMH-CH, relatively few particles are required (ranging from 35–55) even as noise in the observation process reduces. Although PMMH-BPF required fewer particles than PMMH-LF, as σ is reduced, increasing numbers of particles are required by both schemes to optimise overall efficiency. We measure overall efficiency by comparing minimum effective sample size scaled by wall clock time (ESS_{min}/s). When using \mathcal{D}_1 ($\sigma = 10$), there is little difference in overall efficiency between each scheme although PMMH-LF is to be preferred. For dataset \mathcal{D}_2 ($\sigma = 5$), PMMH-BPF and PMMH-LF give comparable performance whilst PMMH-

CH outperforms PMMH-LF by a factors of 2.3. For \mathcal{D}_3 ($\sigma = 1$) PMMH-CH and PMMH-BPF outperform PMMH-LF by factors of 61 and 6.7 respectively. Computational cost precluded the use of PMMH-LF on a dataset with $\sigma < 1$, however, our experiments suggest that PMMH-CH can be successfully applied to synthetic data with $\sigma = 0.1$ by using just $N = 50$ particles. Finally, we note that PMMH-CH appears to outperform PMMH-BPF, and, whereas the latter requires choosing appropriate intermediate resampling times and a tempering parameter γ , PMMH-CH requires minimal tuning. Therefore, in the following example, we focus on the PMMH-CH scheme.

5.3 Motility regulation

We consider here a simplified model of a key cellular decision made by the gram-positive bacterium *Bacillus subtilis* (Sonenshein et al., 2002). This decision is whether or not to grow flagella and become motile (Kearns and Losick, 2005). The *B. subtilis* sigma factor σ^D is key for the regulation of motility. Many of the genes and operons encoding motility-related proteins are governed by this σ factor, and so understanding its regulation is key to understanding the motility decision. The gene for σ^D is embedded in a large operon containing several other motility-related genes, known as the *fla/che* operon. The *fla/che* operon itself is under the control of another σ factor, σ^A , but is also regulated by other proteins. In particular, transcription of the operon is strongly repressed by the protein *CodY*, which is encoded upstream of *fla/che*. *CodY* inhibits transcription by binding to the *fla/che* promoter. Since *CodY* is upregulated in good nutrient conditions, this is thought to be a key mechanism for motility regulation. As previously mentioned, many motility-related genes are under the control of σ^D . For simplicity we focus here on one such gene, *hag*, which encodes the protein *flagellin* (or *Hag*), the key building block of the flagella. It so happens that *hag* is also directly repressed by *CodY*. The regulation structure can be encoded as follows.



Following Wilkinson (2011), we assume that three rate constants are uncertain, namely c_3 (governing the rate of production of SigD), c_9 and c_{10} (governing the rate at which CodY binds or unbinds to the flache promoter). Values of the rate constants are taken to be

$$c = (0.1, 0.0002, 1, 0.0002, 1.0, 0.0002, 0.01, 0.1, 0.02, 0.1, 0.01, 0.1)'$$

and initial values of (codY, CodY, flache, SigD, SigD_hag, hag, Hag, CodY_flache, CodY_hag) are

$$x_0 = (1, 10, 1, 10, 1, 1, 10, 1, 1)'$$

Gillespie's direct method was used to simulate 3 synthetic datasets consisting of 51 observations on SigD only, with inter-observation times of $\Delta t = 1, 2, 5$ time units. A full realisation from the motility model that was used to construct each dataset is shown in Figure 2. The assumed initial conditions and parameter choices give inherently discrete time series.

To provide a challenging (but unrealistic) scenario for the PMMH-CH scheme we assume that error-free observations are available. We adopt independent proper Uniform priors on the log scale:

$$\begin{aligned}
\log(c_3) &\sim \text{U}(\log\{0.01\}, \log\{100\}) \\
\log(c_9) &\sim \text{U}(\log\{0.0002\}, \log\{2\}) \\
\log(c_{10}) &\sim \text{U}(\log\{0.001\}, \log\{10\})
\end{aligned}$$

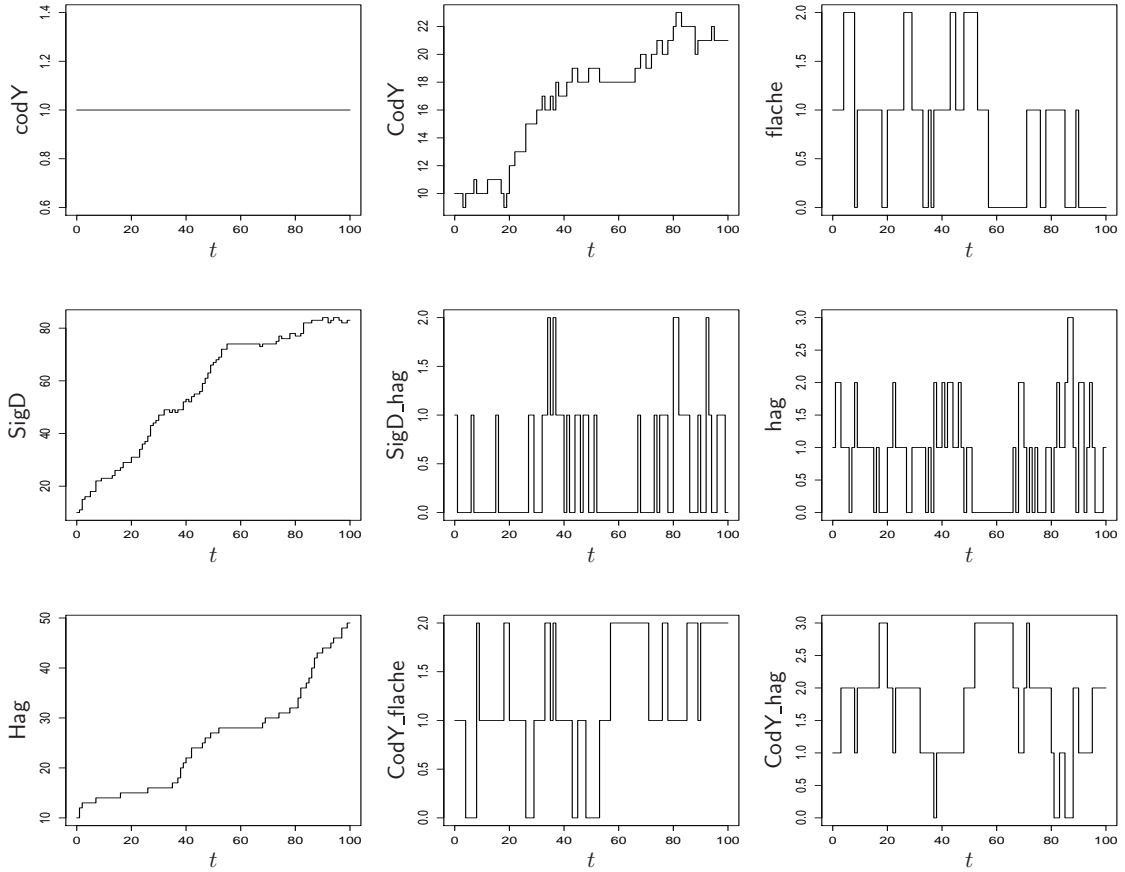


Figure 2: A typical realisation of the motility model.

which cover two orders of magnitude either side of the ground truth. We ran PMMH-CH for 10^5 iterations, after determining (from short pilot runs) suitable numbers of particles for each dataset, and a scaling λ for use in the Gaussian random walk proposal kernel.

Figure 3 shows the marginal posterior distributions for each dataset. We see that despite observing levels of **SigD** only, sampled parameter values are consistent with the ground truth. Table 4 summarises the overall efficiency of PMMH-CH when applied to each dataset. We see that as the inter-observation time Δt increases, larger numbers of particles are required to maintain a variance in log-posterior of around 2 at the estimated posterior median. Despite using increased particle numbers, statistical efficiency, as measured by effective sample size, appears to reduce as Δt is increased. We observed that parameter chains were more likely to “stick” (and note the decreasing acceptance rate) leading to reduced ESS. This is not surprising given the assumptions used to derive the conditioned hazard, and we expect its performance to diminish as inter-observation time increases.

6 Discussion and conclusions

This paper considered the problem of performing inference for the parameters governing Markov jump processes in the presence of informative observations. Whilst it is possible to construct particle MCMC schemes for such models given time course data that may be incomplete and subject to error, the simplest “likelihood-free” implementation is likely to be computationally intractable, except in high measurement error scenarios. To circumvent this issue, we have proposed a novel method for simulating from a conditioned jump process, by approximating the expected number

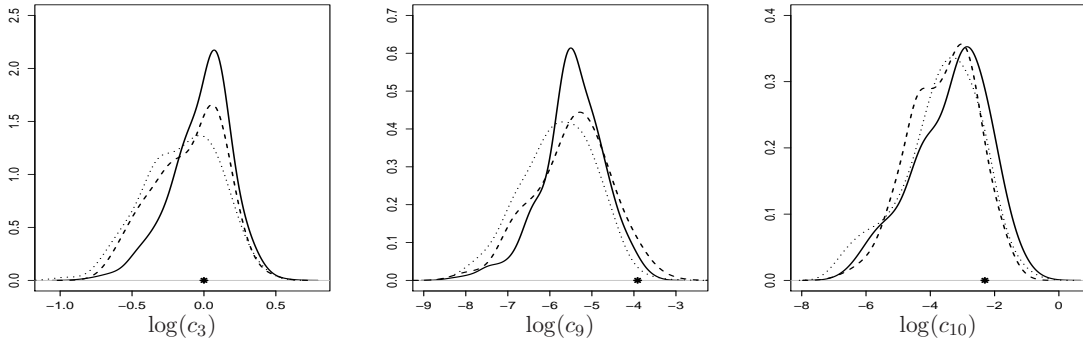


Figure 3: Motility regulation model. Marginal posterior distributions based on synthetic data with inter-observation times of $\Delta t = 1$ (solid), $\Delta t = 2$ (dashed) and $\Delta t = 5$ (dotted). Values of each $\log(c_i)$ that produced the data are indicated.

| | N | τ^2 | Acc. rate | ESS($\theta_1, \theta_2, \theta_3$) | Time (s) | ESS _{min} /s |
|----------------|------|----------|-----------|---------------------------------------|----------|-----------------------|
| $\Delta t = 1$ | 400 | 1.99 | 0.10 | (1635, 2156, 1625) | 6933 | 0.23 |
| $\Delta t = 2$ | 600 | 2.05 | 0.10 | (1870, 1215, 1518) | 6950 | 0.17 |
| $\Delta t = 5$ | 1200 | 2.01 | 0.06 | (797, 791, 673) | 13628 | 0.05 |

Table 4: Motility regulation model. Number of particles N , variance of the noise in the log-posterior (τ^2) at the posterior median, acceptance rate, effective sample size (ESS) of each parameter chain and wall clock time in seconds and minimum (over each parameter chain) ESS per second.

of reactions between observation times to give a conditioned hazard. We find that a simple implementation of this approach, with exponential waiting times between proposed reaction events, works extremely well in a number of scenarios, and even in challenging multivariate settings. It should be noted however, that the assumptions under-pinning the construct are likely to be invalidated as inter-observation time increases. We compared this approach with a bridge particle filter adapted from Del Moral and Murray (2014). Implementation of this approach requires the ability to simulate from the model and access to an approximation of the (unavailable) transition probabilities. The overall efficiency of the scheme depends on the accuracy and computational cost of the approximation. Use of the LNA inside the bridge particle filter appears promising, although the requirement of solving a system of ODEs for each particle, and whose dimension increases quadratically with the number of species, is likely to be a barrier to its successful application in high dimensional systems. Using a numerical approximation to the CLE offers a cheaper but less accurate alternative. The bridge particle filter (based on either the LNA or CLE) requires specification of appropriate intermediate resampling times and, when the approximations are likely to be light tailed relative to the jump process transition probability, a tempering parameter. Use of the conditioned hazard on the other hand requires minimal tuning. This approach was successfully applied to the problem of inferring the rate constants governing a Lotka-Volterra system and a simple model of motility regulation.

Improvements to the proposed algorithms remain of interest and are the subject of ongoing research. For example, when using the bridge particle filter, it may be possible to specify a resampling regime dynamically, based on the expected time to the next reaction event, evaluated at, for example, the LNA mean. An exact implementation of the conditioned hazard approach and the potential improvement it may offer is also of interest, especially for systems with finite state space, which would permit a thinning approach (Lewis and Shedler, 1979) to reaction event simulation.

References

- Andrieu, C., Doucet, A., and Holenstein, R. (2009). Particle Markov chain Monte Carlo for efficient numerical simulation. In L’Ecuyer, P. and Owen, A. B., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 45–60. Springer-Verlag Berlin Heidelberg.
- Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods (with discussion). *J. R. Statist. Soc. B*, 72(3):1–269.
- Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient computation. *Annals of Statistics*, 37:697–725.
- Bailey, N. T. J. (1964). *The elements of stochastic processes with applications to the natural sciences*. Wiley, New York.
- Bailey, N. T. J. (1975). *The mathematical theory of infectious diseases and its applications*. Hafner Press [Macmillan Publishing Co., Inc.], New York, 2nd edition.
- Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160.
- Boys, R. J. and Giles, P. R. (2007). Bayesian inference for stochastic epidemic models with time-inhomogeneous removal rates. *J. Math. Biol.*, 55:223–247.
- Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. L. (2008). Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18:125–135.
- Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York.
- Del Moral, P. and Murray, L. M. (2014). Sequential Monte Carlo with highly informative observations. Available from <http://arxiv.org/abs/1405.4081>.
- Delyon, B. and Hu, Y. (2006). Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and thier Applications*, 116:1660–1675.
- Doucet, A., Pitt, M. K., and Kohn, R. (2014). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. Available from <http://arxiv.org/pdf/1210.1871v3.pdf>.
- Durham, G. B. and Gallant, R. A. (2002). Numerical techniques for maximum likelihood estimation of continuous time diffusion processes. *Journal of Business and Economic Statistics*, 20:279–316.
- Elf, J. and Ehrenberg, M. (2003). Fast evolution of fluctuations in biochemical networks with the linear noise approximation. *Genome Res.*, 13(11):2475–2484.
- Fan, Y. and Shelton, C. R. (2008). Sampling for approximate inference in continuous time Bayesian networks. In *Tenth International Symposium on Artificial Intelligence and Mathematics*.
- Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, 18(2):151–171.
- Fearnhead, P., Giagos, V., and Sherlock, C. (2014). Inference for reaction networks using the Linear Noise Approximation. To appear in *Biometrics*.
- Feller, W. (1952). The parabolic differential equations and the associated semi-groups of transformations. *Annals of Mathematics*, 55:468–519.

- Ferm, L., Lötstedt, P., and Hellander, A. (2008). A hierarchy of approximations of the master equation scaled by a size parameter. *J. Sci. Comput.*, 34(2):127–151.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361.
- Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425.
- Gillespie, D. T. (2000). The chemical Langevin equation. *Journal of Chemical Physics*, 113(1):297–306.
- Golightly, A., Henderson, D. A., and Sherlock, C. (2014). Delayed acceptance particle MCMC for exact inference in stochastic kinetic models. To appear in *Statistics and Computing*.
- Golightly, A. and Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788.
- Golightly, A. and Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, 1(6):807–820.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings-F*, 140:107–113.
- Hajiaghayi, M., Kirkpatrick, B., Wang, L., and Bouchard-Côté, A. (2014). Efficient continuous-time Markov chain estimation. In *31st International Conference on Machine Learning*.
- Kearns, D. B. and Losick, R. (2005). Cell population heterogeneity during growth of *Bacillus subtilis*. *Genes and Development*, 19:3083–3094.
- Komorowski, M., Finkenstadt, B., Harper, C., and Rand, D. (2009). Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics*, 10(1):343.
- Kurtz, T. G. (1970). Solutions of ordinary differential equations as limits of pure jump markov processes. *J. Appl. Probab.*, 7:49–58.
- Lewis, P. A. W. and Shedler, G. S. (1979). Simulation of a nonhomogeneous Poisson process by thinning. *Naval Research Logistics Quarterly*, 26:401–413.
- Lin, M., Chen, R., and Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28:69–94.
- Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90:567–576.
- O’Neill, P. D. and Roberts, G. O. (1999). Bayesian inference for partially observed stochastic epidemics. *J. R. Statist. Soc. A*, 162:121–129.
- Petzold, L. (1983). Automatic selection of methods for solving stiff and non-stiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing*, 4(1):136–148.
- Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econometrics*, 171(2):134–151.
- Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 446:590–599.

- Schauer, M., van der Meulen, F., and van Zanten, H. (2014). Guided proposals for simulating multi-dimensional diffusion bridges. Available from <http://arxiv.org/abs/1311.3606>.
- Sherlock, C., A., G., and Gillespie, C. S. (2014). Bayesian inference for hybrid discrete-continuous systems biology models. To appear in *Inverse Problems*.
- Sherlock, C., Thiery, A., Roberts, G. O., and Rosenthal, J. S. (2013). On the efficiency of pseudo-marginal random walk Metropolis algorithms. Available from <http://arxiv.org/abs/1309.7209>.
- Sonenshein, A. L., A., H. J., and Losick, R., editors (2002). *Bacillus subtilis and its closest relatives*. ASM Press.
- Stramer, O. and Yan, J. (2007). Asymptotics of an efficient Monte Carlo estimation for the transition density of diffusion processes. *Methodology and Computing in Applied Probability*, 9(4):483–496.
- van Kampen, N. G. (2001). *Stochastic Processes in Physics and Chemistry*. North-Holland.
- Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10:122–133.
- Wilkinson, D. J. (2011). Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology (with discussion). In Bernardo, J. M. e. a., editor, *Bayesian Statistics 9*, pages 679–706. OUP.
- Wilkinson, D. J. (2012). *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press, Boca Raton, Florida, 2nd edition.